



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Alternative Approach to Nuclear Data Representation: Building the infrastructure to support QMU and next-generation simulations

J. Pruet, D. A. Brown, B. Beck, D. P. McNabb

January 18, 2006

NEDPC 2005

Livermore, CA, United States

October 17, 2005 through October 21, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Alternative Approach to Nuclear Data Representation: Building the infrastructure to support QMU and next- generation simulations (U)

J. Pruet*, D. Brown*, B. Beck*, D.P. McNabb*

**Lawrence Livermore National Laboratory, Livermore, CA*

The nuclear data infrastructure currently relies on punch-card era formats designed some five decades ago. Though this system has worked well, recent interest in non-traditional and complicated physics processes has demanded a change. Here we present an alternative approach under development at LLNL. In this approach data is described through collections of distinct and self-contained simple data structures. This structure-based format is compared with traditional ENDF and ENDL, which can roughly be characterized as dictionary-based representations.
(U)

Introduction

Simulating nuclear devices requires a mature nuclear data infrastructure. However, collating and quantitatively describing nuclear processes in a coherent way is generally non-trivial. For example, ENDF - the nuclear reaction database and format supporting both the reactor and weapons community - has taken decades of work by many outstanding scientists to develop and maintain.

The data effort is made difficult by a number of factors. At the most basic level, nuclear physics is a rich and complicated subject, with reacting nuclei exhibiting a myriad of resonances, breakups, transitions, etc. Another complication relates to the specific needs of different nuclear data consumers. For example, the need for including resonance parameters becomes clear when one understands details of multi-group transport simulations of various nuclear devices. The last complicating factor is simply dealing with the tools and formats used to represent nuclear data. This is the factor studied here.

In this paper we examine the use of structure-based formats and describe how these formats could benefit the nuclear data infrastructure.

The Need for Change

The term “structure-based” is used here to mean that the format consists of a collection of separate pieces, each of which represents a different idea or data-type. This can be better understood by comparing structure-based formats with more traditional formats currently used for nuclear science.

Current nuclear reaction data storage formats consist roughly of a few simple array-like structures and fixed-formatted character strings. These structures are arranged into files according to well-defined templates. A dictionary (or manual) is used to interpret the data. As an example, ENDL (the data format used at LLNL) consist of a sequence of two-line headers followed by an appropriately sized matrix of data [1]. The 11th and 12th columns in the first row of a header contain a number representing one outgoing particle of the reaction described by the data. Looking in the ENDL manual, we see that a “ 7” in these columns means that the reaction involves an outgoing photon.

As another example, material in ENDF is specified by a “MAT” number. To understand which material a particular MAT number represents, one refers to the ENDF manual [2]:

“...MAT number for isotopes of an element are assigned on the basis of increasing mass in steps of three, allowing for the ground and two meta-stable states ... the lightest stable isotope is assigned the MAT number ZZ25.... For the special case of elements from Es to Lw MAT numbers 99xx are assigned...”

To determine which isotope a MAT number refers to, one needs the table of isotopes and some simple calculations.

The examples above for ENDL and ENDF underscore the remarkable compactness and efficiency of current nuclear formats. Most of the burden of interpretation rests with the dictionary (or manual) rather than with the data file itself. Representing several isomers of every isotope with a single 4 digit number makes very efficient use of memory. This efficiency was essential in the early days of the data infrastructure effort, when punch cards were used. Also, FORTRAN and other languages of the period made implementation of classes or deeply nested structures tedious. This necessitated a limited variety of data structures.

But the storage efficiency and paucity of structures characterizing current nuclear formats comes at a heavy cost. Since the syntax and basic elements comprising these formats are so limited, extension can be ad-hoc and difficult. For example, it isn't clear how to add a fourth isomeric state to targets in ENDF. As another example, to represent fission fragment distributions in ENDL one uses $Z=99$, $A=120$. With enough extensions a format like ENDL eventually devolves into a list of exceptions. Also, it can be difficult to add support for complicated new data types. Our current effort, for example, was motivated by the need to include covariance data and uncertainty data that seem difficult to represent in ENDL.

In addition, maintaining and translating traditional formats is very labor intensive. For example, our computational nuclear physics group has invested several man-years of effort in translating ENDF to ENDL. As another example, an entire community supports ENDF. This support is so challenging that the nuclear data infrastructure currently relies on only one or two central FORTRAN codes at each national laboratory for processing

data and, at the international level, nearly everyone relies on NJOY for nuclear data processing.

An Alternative Approach

An alternative approach to representing nuclear data is one that might be characterized as a structure- or class-based representation. In this approach one first defines a set of simple and general data structures. These will include things like representations of nuclei, levels, vectors, matrix-like objects, and so on. Associations between these structure are used to describe more complicated things like reaction channels. This representation is different from traditional approaches. In ENDL or ENDF, new data types are typically accommodated by adding a new number or word in the data file, along with a corresponding dictionary or manual entry. For example, previously unassigned MAT numbers were designated to represent molecules involved in the coherent scattering of low energy neutrons.

Structure-based representations have a number of advantages over traditional formats. By stacking the building blocks comprising the representation very complicated quantities can be expressed in a straightforward way. As a simple example, one could have a matrix whose elements are themselves arbitrary data structures. The first element might itself be a matrix, the second a simple number, the third an array of arrays, ... Also, the basic data structures (or basis structures) can be built to mirror physics. If one needs to describe reactions involving molecules a structure describing molecules is devised. This might contain an account of the atomic constituents of the molecule, information about vibrational modes and the molecule's mass.

In other words, the workload shifts from trying to pack the data into as small a space as possible into trying to represent the data as faithfully and clearly as possible.

Expressing the Data

Part of designing the new format involves coming up with some convention or syntax for describing the different structures. One mature and widely used solution is XML (eXtensible Markup Language), a convention for describing data stored in tree-like structures. We are adopting XML because it is widely supported, with several options for open source parsers, checkers, etc for a variety of programming languages.

Briefly, XML documents consist of a sequence of nested elements. The beginning of an element represented by a tag "x" is denoted by "<x>" and the end by "</x>". Short elements, with no nested elements can be denoted "<x/>". Elements can contain attributes, which describe simple characteristics of the element, or other elements:

```
<particle particleName="neutron">
    <spin>1/2</spin>
    <mass units="MeV">939.56533</mass>
</particle>
```

Simple data types (strings, integers, etc.), such as particleName above, can receive their own elements or be described as attributes. Each branch on the tree represents a

UNCLASSIFIED

Proceedings of the NEDPC 2005 Document # (e.g. COPD-,COPJ-,etc.)

new instance of some data structure. The mapping of XML elements to classes or data structures in a program is thus straightforward.

For illustration, we present the reaction structure used in LLNL's new format to describe delayed neutrons emitted following $^{238}\text{U}(n,f)$:

```
<reaction>
  <incomingChannel>
    <bodyAccount>
      <multiplicity number='1' qualifier='equal' />
      <particle particleName='neutron' />
    </bodyAccount>
    <bodyAccount>
      <multiplicity number='1' qualifier='equal' />
      <nucleus Z='92' nucleusName='Uranium238' A='238'>
        <excitationState>
          <level>
            <excitationEnergy units='MeV' Val='0.0' />
          </level>
        </excitationState>
      </nucleus>
    </bodyAccount>
    <ambient>
      <Temperature units='MeV' Val='2.586e-08' />
    </ambient>
  </incomingChannel>
  <outgoingChannel>
    <qValue units='MeV' Val='180.0' />
    <intermediateState decayType='weak' stepNumber='1' />
    <bodyAccount>
      <multiplicity number='1' qualifier='equal' />
      <mixture mixtureName='fissionFragment' />
    </bodyAccount>
    <bodyAccount>
      <multiplicity number='1' qualifier='any' />
      <particle particleName='any' />
    </bodyAccount>
    <channelData dataRef='/nuclearData/i.xml' quantityName='nubar' />
  </outgoingChannel>
</reaction>
```

This is interpreted as saying that the incoming channel consists of a single ^{238}U nucleus and a single neutron. The U is in its ground state in equilibrium with a thermal bath at temperature $kT=2.586 \cdot 10^{-8}$ MeV. An intermediate state persists over weak timescale before the final state is reached and before the delayed neutrons are emitted. The outgoing channel is that resulting from a fission reaction. Note the use of the fission fragment `mixture` element in the outgoing channel. This is really just a shorthand for specifying that fission occurred. A full account of the independent and cumulative yields characterizing the fragment distribution would appear somewhere else, and would be cumbersome to include in the channel specification. The `any` occurring in the particle name is used to indicate that the process described is inclusive on all outgoing particles.

Pointwise data describing dependence of the mean number of neutrons emitted as a function of incident neutron energy is contained a separate file (`/nuclear/i.xml` in

UNCLASSIFIED

Proceedings of the NEDPC 2005 Document # (e.g. COPD-,COPJ-,etc.)

this example). This file itself contains a structure-based description of the pointwise data. Included in the description are rules for interpolating on the data, the frame (lab or center of mass) in which different quantities are described and the specific meaning of described quantities.

Where we stand

Structure-based approaches to data representation are characterized by several useful features. Among these are easy extensibility, nearly automatic representation by classes within programs, and the ability to represent complicated data in a natural way. Data formats currently used by the nuclear community can be viewed as consisting of key-dictionary pairs and are relatively weak by these standards. However, dictionary-based formats have the great advantage of compactly representing complicated data. As well, traditional formats have a long, proven record of utility and are widely supported in codes. Whether or not these historical advantages will outweigh advantages of structure-based representations remains to be seen.

Updating the coding infrastructure supporting engineering applications is a greater challenge than designing consistent library standards. Improvements in the way nuclear data is represented are not useful if they break all of the existing application codes. The natural short-term solution, which has been adopted at Livermore, is to maintain translators between structure-based and traditional data representations. With this, none of the processing or application codes need to be changed. In the longer term, we hope that improvements in the data infrastructure will facilitate improvements in the code infrastructure. In particular, a good transparent structure-based representation could bring the job of processing nuclear data within the purview of well-studied code development and object-based programming methods. We feel that this is desirable because the nuclear engineering community currently relies on only one or a very few core codes maintained by a small number of people.

An initial format has been successfully implemented and is now being extensively tested. In addition to representing all of the data described by ENDL, this format also has the capability to represent uncertainties, covariances, and some complicated reactions. We have devised an initial format and developed faithful conversion routines to and from ENDL [3]. We are also trying to get support from the larger community. Our aim is to establish a common central repository of basis structures that different may efforts draw from.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

References

Last name, first and second initial, et al.

5

UNCLASSIFIED

UNCLASSIFIED

Proceedings of the NEDPC 2005 Document # (e.g. COPD-,COPJ-,etc.)

- [1] Howerton, Dye & Perkins, “Evaluated Nuclear Data Library”, LLNL report UCRL-50400 Vol. 4, Rev.1, 1981.
- [2] “ENDF-6 Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VII”, Ed. M. Herman, 2005, BNL-NCS-44945-01/04.
- [3] J. Pruet, D. Brown, B. Beck & D.P. McNabb “Alternative Approach to Nuclear Data Representation”, NIM B, 2005, in press.